# HERMES: KV Cache as Hierarchical Memory for Efficient Streaming Video Understanding

**Haowei Zhang**[1]*, **Shudong Yang**[1,2]*, **Jinlan Fu**[1]†, **See-Kiong Ng**[3], **Xipeng Qiu**[1,2]†

[1]Fudan University, [2]Shanghai Innovation Institute, [3]National University of Singapore
hwzhang25@m.fudan.edu.cn
jinlanjonna@gmail.com, xpqiu@fudan.edu.cn

https://hermes-streaming.github.io/

## Abstract

Recent advancements in Multimodal Large Language Models (MLLMs) have demonstrated significant improvement in offline video understanding. However, extending these capabilities to streaming video inputs, remains challenging, as existing models struggle to simultaneously maintain stable understanding performance, real-time responses, and low GPU memory overhead. To address this challenge, we propose **HERMES**, a novel training-free architecture for real-time and accurate understanding of video streams. Based on a mechanistic attention investigation, we conceptualize KV cache as a hierarchical memory framework that encapsulates video information across multiple granularities. During inference, HERMES reuses a compact KV cache, enabling efficient streaming understanding under resource constraints. Notably, HERMES requires no auxiliary computations upon the arrival of user queries, thereby guaranteeing real-time responses for continuous video stream interactions. HERMES achieves 10× faster TTFT compared to prior SOTA. Even when reducing video tokens by up to 68% compared with uniform sampling, HERMES achieves superior or comparable accuracy across all benchmarks, with up to 11.4% gains on streaming datasets.

## 1 Introduction

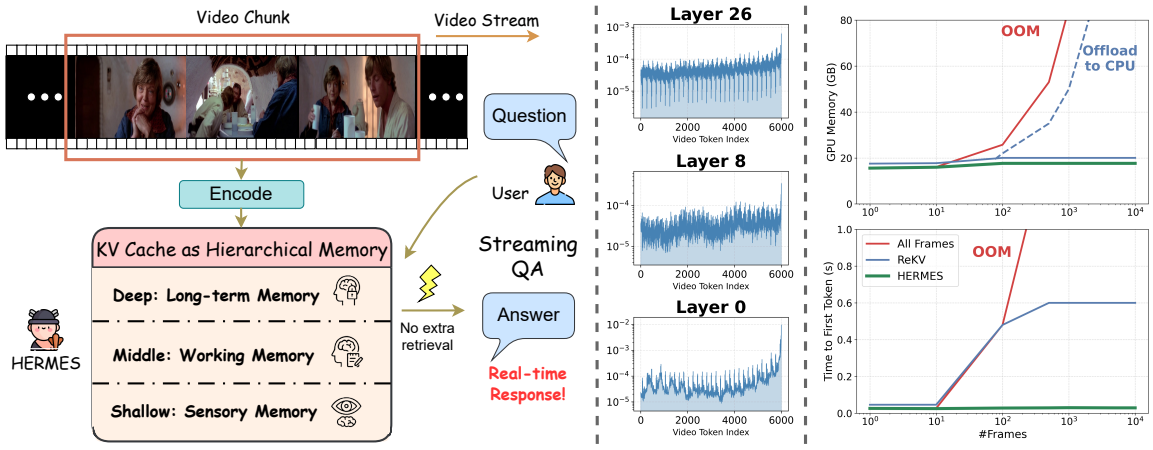
Recent years have witnessed remarkable evolution in the capabilities of Multimodal Large Language Models (MLLMs) in video understanding tasks (Comanici et al., 2025; Li et al., 2024a; Bai et al., 2025a). Despite the progress, the rapid emergence of real-time applications demands stable long video understanding, low-latency response, and memory-efficient deployment. However, existing MLLMs struggle to simultaneously satisfy these requirements on streaming videos. Notably, TimeChat-Online (Yao et al., 2025) observes that a large number of streaming video tokens are redundant, motivating compression methods to address these challenges. While numerous compression techniques have been proposed for offline videos (Wang et al., 2025b; Yang et al., 2024; Tao et al., 2025), most are ill-suited for memory management in streaming scenarios, as streaming inputs are unpredictable in future frames and queries.

To adapt to streaming inputs, recent research introduces specialized memory management techniques, which generally fall into two paradigms: external memory and internal memory. External memory methods store video content as captions or raw vision patches in databases, and perform ad-hoc retrieval and multimodal prefilling at query time (Xiong et al., 2025; Yang et al., 2025a), suffering from high latency and a lack of end-to-end cohesion. Additionally, many of these methods necessitate costly model-specific training (Wang et al., 2025a; Xu et al., 2025; Zeng et al., 2025). In contrast, internalizing memory directly into the key-value cache (KV cache) remains underexplored, yet is crucial for low-latency responses and seamless end-to-end reasoning over stored video contexts. Moreover, KV cache naturally acts as a latent, model-intrinsic memory (Hu et al., 2025) that frequently interacts with the video stream, making it particularly suitable for training-free memory management. ReKV (Di et al., 2025) and LiveVLM (Ning et al., 2025) are representative training-free, cache-based methods for streaming memory management. They store previous video segments in external CPU or disk and need to perform an additional retrieval when a user query arrives, which still rely on external computational resources and leads to significant latency. StreamMem (Yang et al., 2025b) leverages chat template

*Equal Contribution.
†Corresponding authors



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8411–8430
July 2-7, 2026 




(a) HERMES Framework (b) Attention Analysis (c) Efficiency Test

Figure 1: **Left**: HERMES is a training-free approach for efficient streaming video understanding, enabling stable inference by reusing KV cache and performing hierarchical management of video tokens stored in KV cache. **Middle**: HERMES is based on a mechanistic investigation of the layer-wise attention preferences over hierarchical video information. **Right**: We evaluate LLaVA-OV-7B on a single A800 GPU (80 GB). As input frames increase, HERMES consistently maintains extremely low latency (TTFT < 30 ms) and stable GPU memory consumption, exhibiting no risk of OOM errors and requiring no auxiliary external computational resources.

tokens to guide compression but lacks fine-grained KV management and mechanistic interpretability.

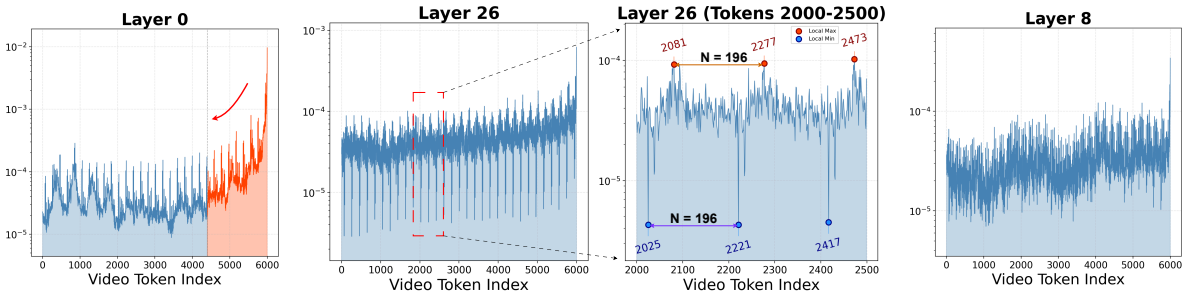
To overcome the aforementioned limitations of existing streaming video methods, we propose **HERMES** (KV Cache as **H**i**ER**archical **M**emory for **E**fficient **S**treaming Video Understanding), a training-free and plug-and-play approach that can be seamlessly integrated into existing MLLMs. Grounded in a mechanistic investigation of layer-wise attention shown in Fig. 1b, we conceptualize KV cache as a hierarchical memory framework that stores video information across multiple levels of granularity: shallow layers function as sensory memory, exhibiting a strong recency bias toward newly arriving frames; deep layers act as long-term memory, focusing on frame-level rhythmic anchor tokens; and middle layers serve as transitional working memory that balances recency information with frame-level semantic representations. Our method HERMES comprises three components: *hierarchical KV cache management*, *cross-layer memory smoothing*, and *position re-indexing*. During inference, HERMES reuses the compact KV cache and requires no auxiliary computations or external devices upon the arrival of user queries, thereby guaranteeing real-time responses. Experiments show that HERMES maintains stable and accurate performance with up to 68% fewer video tokens, while maintaining consistently low response latency and a constant GPU memory footprint.

To summarize, our main contributions are:

1. Grounded in a mechanistic analysis on attention visualization, we pioneer the conceptualization of KV cache as a hierarchical video memory framework across multiple granularities.
2. We propose HERMES, a training-free method for streaming video understanding by reusing hierarchically managed KV cache. Despite reducing video tokens by up to 68%, HERMES achieves competitive accuracy, with gains of up to 11.4% on streaming benchmarks.
3. HERMES exhibits outstanding efficiency in streaming scenarios. Compared to the prior training-free SOTA method, it achieves up to a 10× speedup in latency. With a constant, compact GPU memory footprint and no auxiliary computation at query time, HERMES ensures consistently low-latency responses.

## 2 Layer-wise Preference for Hierarchical Streaming Video Information

Sliding Window is a standard paradigm for streaming video processing by incrementally encoding the continuous video stream chunk by chunk. When KV cache reaches the pre-defined memory budget, token eviction is triggered, and deciding which tokens to keep is crucial for stable understanding. Existing methods (Di et al., 2025; Yang et al., 2025b;




(a) Shallow layer attention. (b) Deep layer attention. (c) Middle layer attention.

Figure 2: Visualization of the average attention weights (log scale) for user queries over video tokens in LLaVA-OV-7B with a FIFO KV cache budget of 6K video tokens per layer, averaged across 300 user video questions.

Xu et al., 2025) rely on coarse-grained eviction strategies such as FIFO uniformly across all layers, overlooking layer-wise attention preferences.

To fill this gap, we conduct a mechanistic investigation of attention preferences in MLLM decoder layers, revealing how layers specialize in storing multiple-granularity video memory. To derive generalized insights, we randomly sample 300 video-question pairs in total from VideoMME benchmark (Fu et al., 2025) to cover diverse video durations and queries. Implementation details of the investigation are provided in App. A. Layer-wise attention visualizations over video tokens maintained in a FIFO KV cache in Fig. 2 reveal three general stages of attention preference, along with more visualization results presented in App. B:

- **Shallow Layers as Sensory Memory**: As shown in Fig. 2a, the shallow layers (e.g., layer 0) exhibit an intense recency bias, with attention sharply concentrated on the most recent visual tokens and rapidly decaying over earlier ones. This behavior aligns with the concept of *Sensory Memory* (Atkinson and Shiffrin, 1968; Shan et al., 2025): shallow layers function as a short-lived buffer for the most recent visual inputs, enabling the model to quickly perceive incoming frames.
- **Deep Layers as Long-term Memory**: In deep layers (e.g., layer 26 in Fig. 2b), recency bias largely disappears. Instead, the attention pattern becomes highly sparse and rhythmic, with local extrema appearing at regular intervals. These extrema are exactly N = 196 tokens apart, matching to the number of tokens encoding a single frame in LLaVA-OV-7B. These local maxima can be regarded as frame-level "anchor tokens", summarizing the visual information of each frame. This pattern reflects *Long-term Memory* (Atkinson and Shiffrin, 1968; Shan et al., 2025): deep layers store critical frame-level semantic representations for long-horizon understanding.
- **Middle Layers as Working Memory**: Middle layers (e.g., layer 8 in Fig. 2c) exhibit a gradual reduction in recency bias, with attention more evenly distributed across recent and earlier tokens. Simultaneously, the attention begins to transition toward the rhythmic patterns in the deep layers. This behavior corresponds to *working memory* (Baddeley and Hitch, 1974; Hu et al., 2025): middle layers integrate recent and earlier visual information, bridging short-term sensory traces with frame-level semantic summaries.

## 3 HERMES

We propose HERMES, a training-free framework that can be seamlessly integrated with MLLMs. As shown in Fig. 3, HERMES has three components: hierarchical KV cache management, cross-layer memory smoothing, and position re-indexing.

### 3.1 Hierarchical KV Cache Management

Motivated by the layer-wise attention patterns identified in Sec. 2, we design a hierarchical KV cache strategy. For each video token with KV cache index $i$ at layer $l$, where $i$ denotes its physical position in KV cache, we compute an importance score $S_i^l$ to decide its retention:

- **Shallow Layers**: They act as sensory memory with strong recency bias. Inspired by Ebbinghaus' memory decay theory (Ebbinghaus, 2013), we model token importance using an exponential forgetting curve based on temporal distance:

$$S_i^l = \alpha_i^l \cdot e^{-k\Delta t_i}, \Delta t_i = T - 1 - i, \quad (1)$$

where $T$ is the total number of video tokens in the cache, $k > 0$ is the forgetting rate, $\alpha_i^l$ denotes the normalization factor.

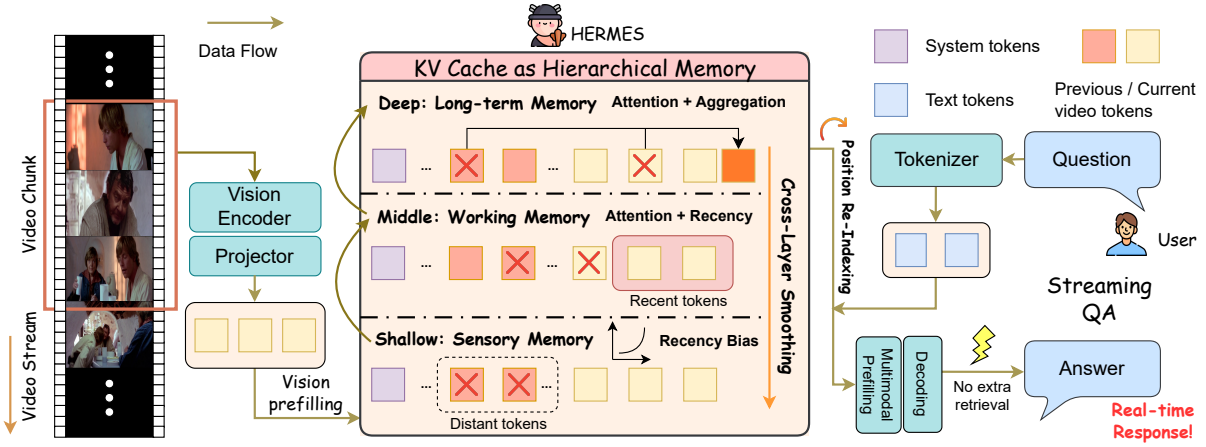



Figure 3: Overview of the HERMES architecture for streaming video QA. By implementing a hierarchical KV cache and specialized management strategies, HERMES enables real-time and accurate responses through direct cache reuse, eliminating the need for additional retrieval operations or external memory whenever users pose questions.

- **Deep Layers**: Deep layers function as frame-level long-term memory with stable anchor tokens. Their attention distributions are sparse, and these anchor tokens consistently receive high attention across frames, making attention magnitude a reliable indicator of long-term importance. We therefore compute token importance directly from attention weights with respect to the user query. To handle unpredictable queries in streaming scenarios, we use a generic guidance prompt (see App. C) as a pseudo query. Token importance is computed as:

$$S_i^l = \alpha_i^l \cdot W_i^l, \tag{2}$$

where $W_i^l$ denotes the attention weight of the $i$-th token at the layer $l$.

- **Middle Layers**: Middle layers serve as working memory, transitioning from recency-dominated shallow layers to attention-driven deep layers. We compute importance by interpolating recency and attention with a layer-dependent weight:

$$\omega^l = \omega_0 - \gamma \cdot \frac{l - l_{\text{short}}}{l_{\text{long}} - l_{\text{short}}}, \tag{3}$$

where $l_{\text{short}}$ and $l_{\text{long}}$ denote the layer indices, with $\omega_0 = 0.75$ and $\gamma = 0.6$. The importance score of token $i$ at layer $l$ is then computed as

$$S_i^l = (1 - \omega^l)\, A_i^l + \omega^l\, R_i^l, \tag{4}$$

where $A_i^l$ and $R_i^l$ denote the normalized attention weight and recency score, respectively, computed as in Eqs. (1) and (2).

### 3.2 Cross-Layer Memory Smoothing

Hierarchical KV cache management may introduce cross-layer inconsistency, as tokens at the same cache index can be evicted independently across layers, leading to misaligned visual memory. Since effective LLM memory relies on cross-layer interaction (Packer et al., 2024; Behrouz et al., 2024; Sun and Zeng, 2025; Hu et al., 2025), we address this issue with *Cross-Layer Memory Smoothing*.

Instead of treating video tokens at the same KV cache index as independent across layers, we propagate and smooth importance signals from deeper to shallower layers. Given raw importance scores $S_i^l$, the smoothed score is computed as:

$$\tilde{S}_i^l = (1 - \lambda_l) \cdot S_i^l + \lambda_l \cdot S_i^{l+1}, \tag{5}$$

$\lambda \in [0, 1]$ is the smoothing hyperparameter that controls the strength of cross-layer smoothing.

We then apply Top-K selection based on $\tilde{S}_i^l$ to maintain a fixed memory budget $|M|$ per layer:

$$\begin{aligned} \mathcal{I}_l &= \text{TopK}(\tilde{S}_l, |M|), \\ K_l &= K_l[\mathcal{I}_l], V_l = V_l[\mathcal{I}_l]. \end{aligned} \tag{6}$$

To preserve long-term information, evicted tokens are aggregated into a ***summary token*** per layer, which compactly encodes long-term memory and is retained in the KV cache (see App. G).

### 3.3 Position Re-Indexing

Continuous accumulation of streaming inputs causes positional indices to exceed the model's maximum supported range, severely degrading text

generation quality. To stabilize inference, we apply position re-indexing, which remaps positional indices to a contiguous range $[0, |M|)$ within the memory budget $|M|$. We design two strategies:

**Lazy Re-Indexing** Position re-indexing is triggered only when positional indices approach the model limit, resulting in lower computational overhead. By preserving the original positional indices of recent tokens, it prevents positional drift compared to eager re-indexing, making it well suited for streaming video understanding.

**Eager Re-Indexing** Re-indexing is performed at each compression step, maintaining strictly contiguous RoPE indices in KV cache. While this strategy stabilizes long-range visual semantics (Kim et al., 2024, 2025; Xu et al., 2025), it leads to higher computational cost due to frequent re-indexing, making it more suitable for offline videos.

The details of re-indexing implementation for 1D RoPE and 3D M-RoPE are illustrated in Sec. F.1 and Sec. F.2, respectively.

# 4 Experiments

## 4.1 Experimental Setup

**Benchmarks.** We evaluate HERMES on diverse streaming and offline benchmarks. For streaming understanding, we use StreamingBench (Lin et al., 2024b), OVO-Bench (Li et al., 2025) and RVS (including RVS-Ego and EVS-Movie) (Zhang et al., 2024a). For offline video evaluation, we adopt one short video dataset MVBench (Li et al., 2024b), along with two long video datasets, VideoMME (Fu et al., 2025) and Egoschema (Mangalam et al., 2023). We conduct evaluation on the official dev split of Egoschema and report VideoMME results without subtitles. Our benchmark selection covers both multiple-choice and open-ended questions as QA form. The details of utilized benchmarks are demonstrated in App. E.

**Models.** To further verify the broad applicability of our method, we select two popular open-source MLLM series, LLaVA-OneVision (LLaVA-OV) (Li et al., 2024a) and Qwen2.5-VL (Bai et al., 2025b). Each is tested across two different parameter scales, covering a large range from 0.5B to 32B. For Qwen2.5-VL, we maintain its native dynamic resolution on video input, ensuring a fair comparison with the base model.

**Implementation Details.** For evaluating HERMES across all benchmarks, each video is encoded and processed chunk by chunk, with 16 frames per chunk, and sequentially prefilling the backbone LLM. Then, token compression is triggered once the predefined memory budget is exceeded.

For the layer partition, we follow the mechanistic investigations presented in Sec. 2: 10% shallow, 60% middle and 30% deep layers. A more comprehensive analysis of attention behaviors as supportive evidence can be found in Fig. 6. The cross-layer memory smoothing hyperparameter $\lambda$ proposed in Sec. 3.2 is layer-dependent, with detailed configurations reported in App. D.

All evaluations are conducted using FP16 mixed precision and efficiency tests are conducted on a single A800 GPU, consistent with prior works (Di et al., 2025; Chen et al., 2025a). Greedy decoding is used to generate deterministic outputs. Accuracy evaluations can be completed on one H200 GPU.

## 4.2 Main Results

**Streaming Video Understanding** Extensive experiments on streaming benchmarks reveal the following key findings:

(1) *HERMES outperforms on multiple-choice streaming datasets, showing exceptional real-time understanding and backward tracing capabilities*. As shown in Tab. 1, it achieves state-of-the-art performance on StreamingBench and OVO-Bench, significantly surpassing base models and training-free baselines. Built on Qwen2.5-VL-7B, HERMES reaches 79.44% and 59.21% accuracy using only 4K video tokens, improving over Qwen2.5-VL-7B by 6.13% and 6.93%, while outperforming all 7B-scale open-source online and offline models. Full results on StreamingBench and OVO-Bench are shown in Tab. 13 and Tab. 14 respectively.

(2) *HERMES excels on open-ended streaming tasks, showing fine-grained temporal and spatial comprehension*. On RVS-Ego and RVS-Movie (Tab. 2), HERMES consistently surpasses all prior training-free methods and improves accuracy by up to 11.4% over the base model with uniformly sampled 64 frames. These extensive experiments demonstrate HERMES's strong abilities in various streaming tasks, as well as its general applicability across foundation models. Moreover, we provide case studies from RVS benchmark, showing finer-grained temporal (shown in Fig. 12) and spatial understanding (shown in Fig. 13) abilities of HERMES than its base model.

Table 1: Performance comparison (%) on StreamingBench and OVO-Bench. The "Avg." column reports the results of the average accuracy of real-time visual perception and backward tracing tasks.

| Model | #Frames | StreamingBench Real-Time | OVO-Bench Real-Time | OVO-Bench Backward | Avg. |
|---|---|---|---|---|---|
| **Open-source Offline MLLMs** | | | | | |
| Video-LLaMA2-7B (Cheng et al., 2024) | 32 | 49.52 | - | - | - |
| VILA-1.5-8B (Lin et al., 2024a) | 14 | 52.32 | - | - | - |
| Video-CCAM-14B (Fei et al., 2024) | 96 | 53.96 | - | - | - |
| LongVA-7B (Zhang et al., 2024c) | 128 | 59.96 | - | - | - |
| Qwen2-VL-7B (Wang et al., 2024b) | 64 | 69.04 | 60.65 | 48.58 | 54.62 |
| InternVL-V2-8B (Chen et al., 2024b) | 16 | 63.72 | 60.73 | 44.00 | 52.37 |
| LLaVA-NeXT-Video-32B (Liu et al., 2024a) | 64 | 66.96 | - | - | - |
| MiniCPM-V-2.6-8B (Hu et al., 2024) | 32 | 67.44 | - | - | - |
| **Open-source Online MLLMs** | | | | | |
| Flash-VStream-7B (Zhang et al., 2024b) | - | 23.23 | 29.86 | 25.35 | 27.61 |
| VideoLLM-online-8B (Chen et al., 2024a) | 2 fps | 35.99 | 20.79 | 17.73 | 19.26 |
| Dispider-7B (Qian et al., 2025) | 1 fps | 67.63 | 54.55 | 36.06 | 45.31 |
| TimeChat-Online-7B (Yao et al., 2025) | 1 fps | 75.36 | 61.90 | 41.70 | 51.80 |
| StreamForest-7B (Zeng et al., 2025) | 1 fps | 77.26 | 61.20 | 52.02 | 56.61 |
| **Training-free Offline-to-Online Methods** | | | | | |
| LLaVA-OV-7B (Li et al., 2024a) | 64 | 71.34 | 63.06 | 43.64 | 53.35 |
| + ReKV (Di et al., 2025) | 0.5 fps | 69.22 | 57.33 | 44.16 | 50.75 |
| + LiveVLM (Ning et al., 2025) | 0.5 fps | 72.92 | - | - | - |
| + StreamKV (Chen et al., 2025b) | 0.5 fps | 68.80 | - | - | - |
| + HERMES (6K tokens) | 0.5 fps | 72.63 | 65.07 | 48.80 | 56.94 |
| + HERMES (4K tokens) | 0.5 fps | **73.23** | **66.34** | **50.20** | **58.27** |
| LLaVA-OV-0.5B (Li et al., 2024a) | 64 | 59.64 | 49.70 | 34.59 | 42.15 |
| + ReKV (Di et al., 2025) | 0.5 fps | 57.39 | 43.77 | 33.06 | 38.42 |
| + HERMES (6K tokens) | 0.5 fps | 61.04 | 50.34 | 34.75 | 42.55 |
| + HERMES (4K tokens) | 0.5 fps | **62.04** | **50.72** | **34.80** | **42.76** |
| Qwen2.5-VL-7B (Bai et al., 2025b) | 1 fps | 73.31 | 59.90 | 44.65 | 52.28 |
| + HERMES (6K tokens) | 1 fps | 78.72 | 68.42 | 48.10 | 58.26 |
| + HERMES (4K tokens) | 1 fps | **79.44** | **68.98** | **49.43** | **59.21** |
| Qwen2.5-VL-32B (Bai et al., 2025b) | 1 fps | 74.27 | 64.40 | 50.33 | 57.37 |
| + HERMES (6K tokens) | 1 fps | **80.20** | 71.93 | **57.71** | **64.82** |
| + HERMES (4K tokens) | 1 fps | 80.08 | **72.37** | 55.42 | 63.90 |
| Qwen3-VL-8B (Bai et al., 2025a) | 2 fps | 78.92 | 68.64 | 47.03 | 57.84 |
| + HERMES (6K tokens) | 2 fps | **81.32** | 73.21 | 46.78 | 60.00 |
| + HERMES (4K tokens) | 2 fps | 81.28 | **73.29** | **49.28** | **61.29** |
| Qwen3-VL-4B (Bai et al., 2025a) | 2 fps | 78.32 | 70.67 | 50.05 | 60.36 |
| + HERMES (6K tokens) | 2 fps | **78.40** | 71.90 | 54.00 | 62.95 |
| + HERMES (4K tokens) | 2 fps | 78.24 | **72.32** | **55.03** | **63.68** |

**Offline Video Understanding** The results presented in Tab. 3 demonstrate the *competitive performance of HERMES across multiple temporal scales on offline benchmarks*, compared to the base model and other training-free methods. Under a limited budget of video tokens, HERMES achieves performance that is better than or comparable to the corresponding base models. HERMES based on LLaVA-OV-7B surpasses the base model on long video datasets Egoschema and VideoMME, achieving 60.29% and 58.85%, respectively, and attains 56.92% accuracy on the short video dataset MVBench, which is comparable to the base model's 57.02%.

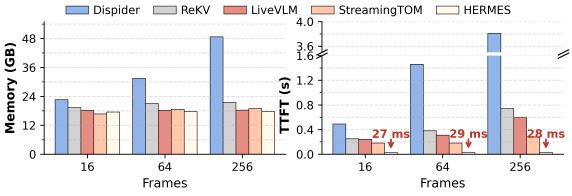



Figure 4: GPU memory and TTFT latency comparison across input frame numbers. HERMES achieves 10 × faster in TTFT compared to prior SOTA.

### 4.3 Efficiency Analysis

To evaluate the efficiency of HERMES, we utilize three metrics: peak GPU memory usage, Time to First Token (TTFT) and Time Per Output Token (TPOT) across varying numbers of input frames.

Table 2: Performance on RVS-Ego and RVS-Movie. "Acc" denotes accuracy (%), and "Score" indicates response quality rated by GPT-3.5-turbo-0125 on a 1–5 scale (consistent with compared baselines). †: ReKV caches the KV states of all previously seen frames and is therefore treated as an upper bound.

| Model | RVS-Ego | | RVS-Movie | |
|---|---|---|---|---|
| | Acc | Score | Acc | Score |
| LLaVA-OV-7B (Li et al., 2024a) | 56.2 | 3.7 | 43.0 | 3.3 |
| + ReKV† (Di et al., 2025) | 63.7 | 4.0 | 54.4 | 3.6 |
| + ReKV w/o off. (Di et al., 2025) | 55.8 | 3.3 | 50.8 | 3.4 |
| + Flash-VStream (Zhang et al., 2024b) | 57.0 | 4.0 | 53.1 | 3.3 |
| + InfiniPot-V (Kim et al., 2025) | 57.9 | 3.5 | 51.4 | 3.5 |
| + StreamMem (Yang et al., 2025b) | 57.6 | 3.8 | 52.7 | 3.4 |
| + StreamingTOM (Chen et al., 2025a) | 58.3 | 3.9 | 53.2 | 3.5 |
| + HERMES (6K tokens) | **60.3** | **4.0** | **54.4** | **3.6** |
| + HERMES (4K tokens) | 58.3 | 3.9 | **54.4** | **3.6** |
| LLaVA-OV-0.5B (Li et al., 2024a) | 51.8 | 3.7 | 37.2 | 3.2 |
| + ReKV† (Di et al., 2025) | 54.7 | 3.9 | 44.6 | 3.4 |
| + HERMES (6K tokens) | **53.0** | **3.8** | **42.5** | **3.4** |
| + HERMES (4K tokens) | 52.7 | **3.8** | 41.7 | **3.4** |

Table 3: Performance comparison (%) on offline benchmarks including MVBench, Egoschema and VideoMME (w/o subtitles). MV: MVBench; Ego: Egoschema; VMME: VideoMME.

| Model | #F | MV | Ego | VMME | |
|---|---|---|---|---|---|
| | | | | Long | Avg. |
| **Open-source Offline MLLMs** | | | | | |
| LLaVA-Video-7B (Zhang et al., 2025) | 32 | 58.60 | 57.30 | - | 63.30 |
| Qwen2-VL-7B (Wang et al., 2024b) | 64 | 67.00 | 66.70 | - | 63.30 |
| InternVL-V2-8B (Chen et al., 2024b) | 16 | 65.80 | - | - | 56.30 |
| **Open-source Online MLLMs** | | | | | |
| Dispider-7B (Qian et al., 2025) | 1 fps | - | 55.60 | - | 57.20 |
| TimeChat-Online-7B (Yao et al., 2025) | 1 fps | 75.36 | 61.90 | 41.70 | 53.22 |
| StreamForest-7B (Zeng et al., 2025) | 1 fps | 70.20 | - | - | 61.40 |
| **Training-free Offline-to-Online Methods** | | | | | |
| LLaVA-OV-7B (Li et al., 2024a) | 64 | **57.02** | 59.93 | 48.00 | 57.67 |
| + ReKV (Di et al., 2025) | 0.5 fps | 56.83 | **60.70** | 46.89 | 57.74 |
| + HERMES (6K tokens) | 0.5 fps | 56.95 | 60.23 | 49.11 | 58.44 |
| + HERMES (4K tokens) | 0.5 fps | 56.92 | 60.29 | **49.22** | **58.85** |
| Qwen2.5-VL-7B (Bai et al., 2025b) | 1 fps | 65.00 | 58.47 | 53.89 | **64.52** |
| + HERMES (6K tokens) | 1 fps | 65.40 | 59.47 | **54.44** | 62.00 |
| + HERMES (4K tokens) | 1 fps | **65.53** | **59.97** | 53.44 | 60.63 |

Table 4: Efficiency across input frame numbers under two chunk sizes. "TTFT" denotes *Time to First Token* and "TPOT" denotes *Time Per Output Token*.

| Metric | Frames | | | |
|---|---|---|---|---|
| | 16 | 64 | 256 | 512 |
| ***Chunk Size: 8*** | | | | |
| GPU Mem. / GB ↓ | 16.54 | 16.66 | 16.66 | 16.66 |
| TTFT / ms ↓ | 27.01 | 28.41 | 28.44 | 28.41 |
| TPOT / ms ↓ | 24.43 | 23.89 | 24.02 | 23.98 |
| ***Chunk Size: 16*** | | | | |
| GPU Mem. / GB ↓ | 17.46 | 17.66 | 17.66 | 17.66 |
| TTFT / ms ↓ | 27.02 | 28.97 | 28.50 | 28.38 |
| TPOT / ms ↓ | 24.50 | 23.59 | 23.56 | 23.63 |

All experiments are conducted using LLaVA-OV-7B as the foundation model with a 4K-token memory budget. Fig. 4 shows the comparison of memory usage and TTFT among HERMES and representative streaming methods. Unlike Dispider and LiveVLM, HERMES consistently maintains stable memory usage and TTFT as frames increase. Notably, under the 256-frame setting, HERMES achieves 1.04× reduction in peak GPU memory usage compared to the prior SOTA LiveVLM, while achieving an impressive 10× speedup in TTFT over the prior SOTA StreamingTOM.

We further examine the efficiency of HERMES under varying encoded video chunk sizes, with the results shown in Tab. 4. GPU memory usage does not increase with longer video lengths due to the fixed memory budget. TTFT and TPOT remain consistently low across varying video lengths and encoding chunk sizes, confirming real-time responsiveness in practical streaming scenarios.

### 4.4 Ablation Study

We conduct ablation studies to evaluate the contributions of HERMES's components and hyperparameter choices, covering: (1) total memory budget, (2) layer-dependent memory budget (3) cross-layer memory smoothing and its hyperparameters, (4) position re-indexing strategies for streaming and offline datasets, (5) guidance prompts and (6) summary tokens for long-term memory retention.

**Total Memory Budget** To investigate the impact of total memory budget on understanding performance, we conduct ablations by varying the memory budget $|M|$ from 1K to 10K. As shown in Fig. 5, for HERMES built upon LLaVA-OV-7B, the performance on both streaming and offline datasets stabilizes once memory budget reaches 4K. Notably, streaming datasets can tolerate a smaller memory budget. In contrast, the performance on long offline datasets degrades significantly when the memory budget is below 4K. The additional ablation on Qwen2.5-VL-7B is provided in App. H, yielding conclusions consistent with those on LLaVA-OV-7B.

**Layer-dependent Memory Budget** We conduct an ablation study on layer-dependent budgets, where the total token budget remains fixed. The allocation strategy for layer-dependent budgets is described as follows: Given a fixed total memory budget $|M| = 6000 \times L$, we allocate per-layer bud-

Table 5: Ablation on layer-dependent budgets.

| Budget Weight | | | StreamingBench | OVO-Bench | | | VideoMME | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Shallow | Middle | Deep | Real-Time | Real-Time | Backward | **Avg.** | Short | Medium | Long | **Avg.** |
| 1.0 | 1.0 | 1.0 | 72.63 | 65.07 | 48.80 | 56.94 | 71.33 | 54.89 | 49.11 | 58.44 |
| 1.3 | 1.0 | 0.7 | 72.42 | 65.28 | **49.27** | **57.28** | 70.78 | 53.11 | 48.89 | 57.59 |
| 0.7 | 1.6 | 0.7 | **72.95** | 64.63 | 48.17 | 56.40 | 70.67 | 54.11 | 47.67 | 57.48 |
| 0.7 | 1.0 | 1.3 | 72.79 | **65.38** | 48.62 | 57.00 | **71.44** | **56.00** | **49.78** | **59.07** |

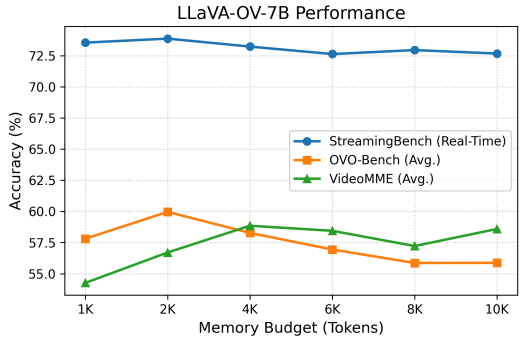



Figure 5: Performance comparison of LLaVA-OV-7B across different memory budgets.

gets proportionally to normalized weights: $m_i = \left\lfloor \frac{w_i}{\sum_j w_j} |M| \right\rfloor$. The rounding residue $|M| - \sum_i m_i$ is added to the last layer to ensure $\sum_i m_i = |M|$, where $w_i$ is the budget weight of the $i$-th layer, $L$ is the number of layers. The results in Tab. 5 show comparable overall performance across configurations, indicating that HERMES is not highly sensitive to the exact layer-dependent budget allocation strategy. Notably, allocating more tokens to deep layers leads to better preservation of long-term memory and improved performance on the long video subset of VideoMME, which is consistent with our observation on layer-wise attention.

Table 6: Ablation on different cross-layer memory smoothing hyperparameter $\lambda$.

| Smoothing Hyperparameter | | | VideoMME | | | |
|---|---|---|---|---|---|---|
| $\lambda_{deep}$ | $\lambda_{mid}$ | $\lambda_{shallow}$ | Short | Medium | Long | **Avg.** |
| 0 | 0 | 0 | 69.67 | 51.11 | 43.44 | 54.74 |
| 0.5 | 0 | 0 | 69.67 | 51.44 | 43.56 | 54.89 |
| 0 | 0.5 | 0 | 70.89 | 54.78 | 46.44 | 57.37 |
| 0 | 0 | 0.5 | 70.89 | 54.44 | 47.00 | 57.44 |
| 0.5 | 0.5 | 0.5 | **71.78** | 54.78 | 47.33 | 57.96 |
| 0.4 | 0.3 | 0.1 | 71.33 | **54.89** | **49.11** | **58.44** |

**Cross-Layer Memory Smoothing** In Tab. 6, we evaluate variants without the proposed cross-layer memory smoothing mechanism, as well as alternative hyperparameter configurations. All these variants exhibit degraded performance on the VideoMME benchmark, demonstrating both the critical role of memory smoothing and the effectiveness of our chosen hyperparameter settings.

Table 7: Ablation on different re-indexing strategies on streaming benchmarks. *"ReI"* refers to "Re-Indexing".

| Model | ReI | StreamBench | OVO-Bench | | |
|---|---|---|---|---|---|
| | | Real-Time | Real-Time | Backward | **Avg.** |
| LLaVA-OV-7B | - | 71.34 | 63.06 | 43.64 | 53.35 |
| + HERMES | lazy | **72.63** | **65.07** | **48.80** | **56.94** |
| + HERMES | eager | 72.30 | 64.91 | 47.21 | 56.06 |

Table 8: Ablation on different re-indexing strategies on VideoMME (offline).

| Model | Re-Indexing | VideoMME | | | |
|---|---|---|---|---|---|
| | | Short | Medium | Long | **Avg.** |
| LLaVA-OV-7B | - | 69.89 | 55.11 | 48.00 | 57.67 |
| + HERMES | lazy | 69.67 | 51.67 | 43.44 | 54.93 |
| + HERMES | eager | **71.33** | 54.89 | **49.11** | **58.44** |

**Position Re-Indexing Strategies** For all streaming evaluations, we adopt the lazy position re-indexing strategy, while we use the eager re-indexing strategy for offline evaluations. Ablation studies in Tab. 7 and Tab. 8 show the effectiveness of these strategies in their respective scenarios.

**Guidance Prompts** To verify that the effectiveness of the token eviction strategy does not depend on a specific prompt design, we conduct an ablation study using three alternative guidance prompts. The results in Tab. 9 show consistent performance across prompt variations, indicating that the method is largely insensitive to the exact wording or design of the guidance prompt.

**Summary Tokens in Deep Layers** In Sec. 3.2, we aggregate the evicted tokens in each deep layer into one summary token at each compression step. The results in Tab. 10 indicate that these summary tokens effectively preserve long-term memory, leading to improved performance on VideoMME.

Table 9: Ablation on guidance prompts. The "generic prompt" refers to the guidance prompt utilized in the paper.

| Guidance Prompt | StreamingBench | OVO-Bench | | | VideoMME | | | |
|---|---|---|---|---|---|---|---|---|
| | Real-Time | Real-Time | Backward | **Avg.** | Short | Medium | Long | **Avg.** |
| **HERMES based on LLaVA-OV-7B** | | | | | | | | |
| generic prompt | 72.63 | 65.07 | 48.80 | 56.94 | **71.33** | **54.89** | **49.11** | **58.44** |
| "What happens in the video?" | **72.75** | **65.49** | 48.60 | 57.05 | 71.11 | 54.11 | 47.67 | 57.63 |
| "Describe the video in detail." | 72.71 | 65.39 | **49.48** | **57.44** | 70.33 | 53.44 | 47.78 | 57.19 |
| "Summarize the content of the video." | 72.55 | 65.45 | 49.19 | 57.32 | **71.33** | 53.00 | 48.22 | 57.52 |
| **HERMES based on Qwen2.5-VL-7B** | | | | | | | | |
| generic prompt | 78.72 | 68.42 | 48.10 | 58.26 | **70.44** | **61.11** | **54.44** | **62.00** |
| "What happens in the video?" | 78.84 | **69.40** | **49.36** | **59.38** | 70.11 | 59.33 | 53.22 | 60.89 |
| "Describe the video in detail." | 78.92 | 68.90 | 49.13 | 59.02 | 70.33 | 59.78 | 53.78 | 61.30 |
| "Summarize the content of the video." | **79.00** | 68.95 | 49.14 | 59.05 | 70.33 | 59.67 | **54.44** | 61.48 |

Table 10: Ablation on summary tokens in deep layers. The gray row is our default setting in all experiments.

| Model | Setting | VideoMME | | | |
|---|---|---|---|---|---|
| | | Short | Medium | Long | **Avg.** |
| LLaVA-OV-7B | - | 69.89 | **55.11** | 48.00 | 57.67 |
| + HERMES | no summary tokens | **71.33** | 54.78 | 47.78 | 57.96 |
| + HERMES | with summary tokens | **71.33** | 54.89 | **49.11** | **58.44** |

## 5 Related Work

**Streaming Video Understanding** Existing MLLMs (Comanici et al., 2025; Li et al., 2024a; Bai et al., 2025b,a) are primarily designed for pre-defined offline videos and struggle with continuous streaming videos. While some prior works have adapted existing offline MLLMs to online settings (Yao et al., 2025; Zeng et al., 2025; Xu et al., 2025), they rely on costly model-specific training. Training-free streaming methods, such as ReKV (Di et al., 2025) and LiveVLM (Ning et al., 2025), prefill offload KV cache to external devices. At user query time, they retrieve the full KV cache and reconstruct it on the GPU, incurring high latency and overall memory usage. In contrast, StreamMem (Yang et al., 2025b) heuristically reuses KV cache, but lacks fine-grained KV cache management and interpretability. Unlike prior training-free methods, HERMES is grounded in a systematic attention analysis with improved interpretability and reliability.

**KV Cache Compression for Video Input** Numerous KV cache compression techniques have been proposed for offline video understanding (Yang et al., 2024; Wang et al., 2024a, 2025b; Tao et al., 2025), but most of these methods are poorly suited for streaming scenarios due to the unpredictable future frames and user queries (Chen et al., 2025a). Existing online KV cache compression paradigms(Di et al., 2025; Ning et al., 2025; Yang et al., 2025b; Chen et al., 2025a) largely overlook the inherently hierarchical storage structure of the KV cache. HERMES addresses this gap by introducing a hierarchical KV cache management strategy, which enables fine-grained memory utilization and low-latency responses.

## 6 Conclusion

This paper proposes HERMES, a training-free framework for efficient streaming video understanding. Guided by mechanistic attention analysis, we conceptualizes KV cache as a hierarchical video memory system across multiple granularities. By introducing a cross-layer memory smoothing and position re-indexing, HERMES further enhances the understanding performance for long streaming input. Extensive experiments demonstrate that HERMES delivers accurate performance under continuously growing video streams, while consistently maintaining extremely low response latency and compact GPU memory usage, making it well suited for real-world streaming deployment.

## Limitations

While our evaluations have spanned a diverse range of MLLMs, due to computation resource constraints, we are unable to implement experiments on the 72B variant (e.g., Qwen2.5-VL-72B). Additionally, we do not investigate the integration of our method with other orthogonal training-free techniques, which may further enhance both understanding performance and efficiency of MLLMs in streaming video scenarios. We plan to conduct more extensive validation involving larger-scale

MLLMs as computational overhead permits.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. U24B20181 and 62521004). This research/project is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

## References

Anthropic. 2024. Claude 3.5 sonnet.

R.C. Atkinson and R.M. Shiffrin. 1968. Human memory: A proposed system and its control processes.

Alan D. Baddeley and Graham Hitch. 1974. Working memory.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. 2024. Titans: Learning to memorize at test time. *Preprint*, arXiv:2501.00663.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 2024a. Videollm-online: Online video large language model for streaming video. *Preprint*, arXiv:2406.11816.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. 2025a. Streamingtom: Streaming token compression for efficient video understanding. *Preprint*, arXiv:2510.18269.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, Ming Lu, and Shanghang Zhang. 2025b. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. *Preprint*, arXiv:2511.07278.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, and 16 others. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Preprint*, arXiv:2404.16821.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *Preprint*, arXiv:2406.07476.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. 2025. Streaming video question-answering with in-context video kv-cache retrieval. *Preprint*, arXiv:2503.00540.

Hermann Ebbinghaus. 2013. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155.

Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. 2024. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *Preprint*, arXiv:2408.14023.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *Preprint*, arXiv:2405.21075.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. *Preprint*, arXiv:2110.07058.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng

Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, and 6 others. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *Preprint*, arXiv:2404.06395.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. Memory in the age of ai agents. *Preprint*, arXiv:2512.13564.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. Movienet: A holistic dataset for movie understanding. *Preprint*, arXiv:2007.10937.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. 2024. Infinipot: Infinite context processing on memory-constrained llms. *Preprint*, arXiv:2410.01518.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. 2025. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *Preprint*, arXiv:2506.15745.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. *Preprint*, arXiv:2311.17005.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. 2025. Ovo-bench: How far is your video-llms from real-world online video understanding? *Preprint*, arXiv:2501.05510.

Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. 2024a. Vila: On pre-training for visual language models. *Preprint*, arXiv:2312.07533.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024b. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *Preprint*, arXiv:2411.03628.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. 2024b. Kangaroo: A powerful video-language model supporting long-context video input. *Preprint*, arXiv:2408.15542.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Preprint*, arXiv:2308.09126.

Zhenyu Ning, Guangda Liu, Qihao Jin, Wenchao Ding, Minyi Guo, and Jieru Zhao. 2025. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. *Preprint*, arXiv:2505.15269.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, and 5 others. 2023. Perception test: A diagnostic benchmark for multimodal video models. *Preprint*, arXiv:2305.13786.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. 2025. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. *Preprint*, arXiv:2501.03218.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. Ntu rgb+d: A large scale dataset for 3d human activity analysis. *Preprint*, arXiv:1604.02808.

Lianlei Shan, Shixian Luo, Zezhou Zhu, Yu Yuan, and Yong Wu. 2025. Cognitive memory in large language models. *Preprint*, arXiv:2504.02441.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. 2024. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *Preprint*, arXiv:2410.17434.

Haoran Sun and Shaoning Zeng. 2025. Hierarchical memory for high-efficiency long-term reasoning in llm agents. *Preprint*, arXiv:2507.22925.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. *Preprint*, arXiv:2411.15024.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. 2025a. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *Preprint*, arXiv:2505.05467.

Han Wang, Yuxiang Nie, Yongjie Ye, Deng GuanYu, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. 2024a. Dynamic-vlm: Simple dynamic visual token compression for videollm. *Preprint*, arXiv:2412.09530.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2025b. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *Preprint*, arXiv:2405.19209.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. 2025. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *Preprint*, arXiv:2501.13468.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. 2025. Streamingvlm: Real-time understanding for infinite video streams. *Preprint*, arXiv:2510.09608.

Haolin Yang, Feilong Tang, Lingxiao Zhao, Xiang An, Ming Hu, Huifa Li, Xinlin Zhuang, Yifan Lu, Xiaofeng Zhang, Abdalla Swikir, Junjun He, Zongyuan Ge, and Imran Razzak. 2025a. Streamagent: Towards anticipatory agents for streaming video understanding. *Preprint*, arXiv:2508.01875.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. Visionzip: Longer is better but not necessary in vision language models. *Preprint*, arXiv:2412.04467.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. 2025b. Streammem: Query-agnostic kv cache memory for streaming video understanding. *Preprint*, arXiv:2508.15717.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. 2025. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. *Preprint*, arXiv:2504.17343.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. 2025. Streamforest: Efficient online video understanding with persistent event memory. *Preprint*, arXiv:2509.24871.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024a. Flashvstream: Memory-based real-time understanding for long video streams. *Preprint*, arXiv:2406.08085.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024b. Flashvstream: Memory-based real-time understanding for long video streams. *Preprint*, arXiv:2406.08085.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024c. Long context transfer from language to vision. *Preprint*, arXiv:2406.16852.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2025. Llava-video: Video instruction tuning with synthetic data. *Preprint*, arXiv:2410.02713.

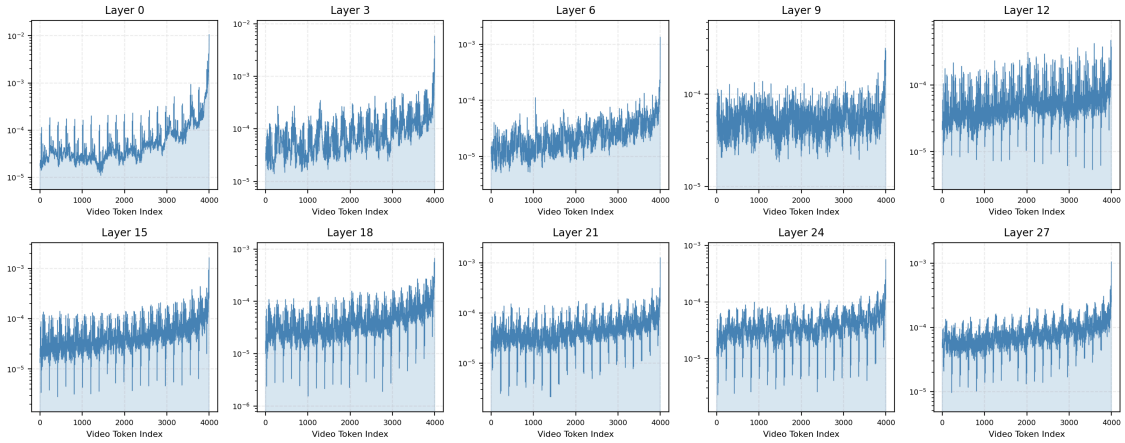
## A Implementation Details of Attention Investigation

During the mechanistic investigation of layer-wise attention preferences in Sec. 2, to extract more general insights, we randomly sample 100 video-question pairs from each of the short (62s[1] - 141s), medium (251s - 1,092s) and long (1,795s - 3,579s) duration subsets of the VideoMME benchmark (Fu et al., 2025) to cover diverse video durations and user queries. The video samples are uniformly sampled at 0.5 fps and subsequently fed into LLaVA-OV-7B in a streaming chunk-wise manner, with each chunk containing 8 frames. LLaVA-OV-7B consists of 28 decoder layers, and each video frame is uniformly encoded into 196 visual tokens. During the prefilling stage for video tokens, we maintain a constant budget $|M|$ of 6K video tokens per KV cache layer. After each eviction step, the positional indices of tokens per KV cache layer are re-indexing to contiguous [0, $|M|$).

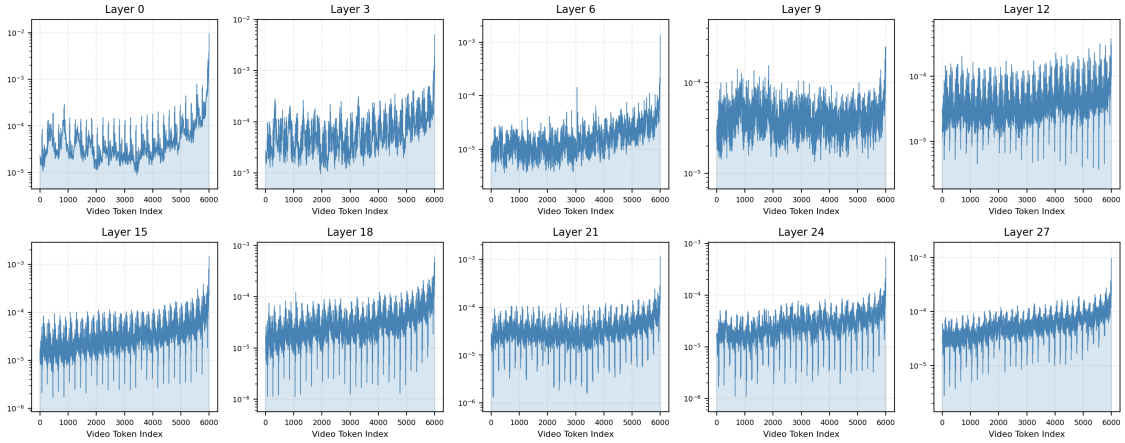
## B More Attention Visualization

We provide more detailed attention visualization in Fig. 6 under different sliding window sizes, showing that the observed attention patterns consistently hold across varying window lengths, thus confirming the generality of the findings in Sec. 2.

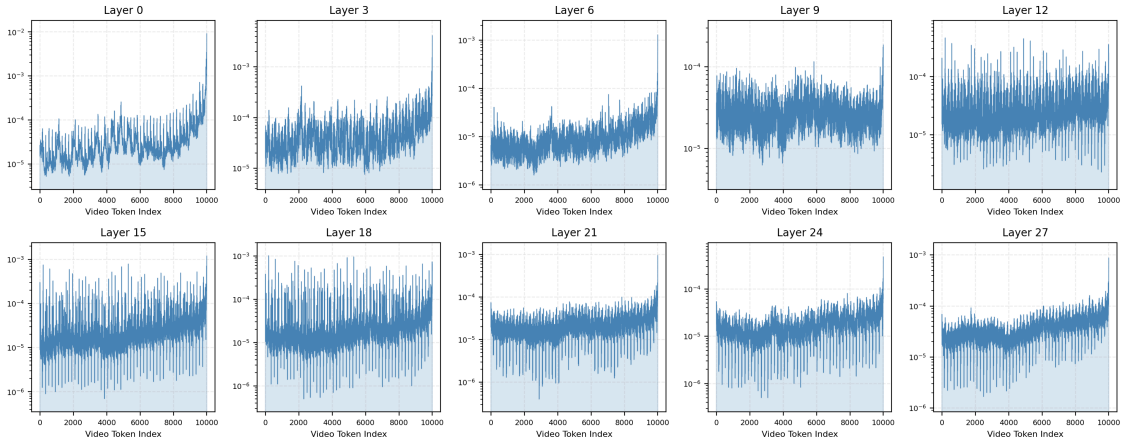
[1]To ensure the sliding window contains 6,000 tokens, a video at 0.5 fps for LLaVA-OV must have a duration of at least $6,000/196/0.5 \approx 62s$.




(a) Sliding window of 4,000 video tokens




(b) Sliding window of 6,000 video tokens




(c) Sliding window of 10,000 video tokens

Figure 6: Visualization of the average attention weights of video tokens in LLaVA-OV-7B under different sliding window sizes.

## C Guidance Prompt

The following two figures show the local and global guidance prompt with and without conversation history to guide the token compression, respectively. For the deep layers, since they primarily focus on frame-level global semantic information, we employ a global guidance prompt as a pseudo-query to extract attention weights of video tokens. In contrast, the middle layers lie in a transition between recency-biased attention and global semantic focus. Therefore, we adopt a hybrid guidance strategy, in which the local guidance prompt and the global guidance prompt are concatenated into a single prompt string to jointly guide the token compression.

> Find recent details related to: {last_conv}. Describe the current scene in detail, focusing on specific objects, fine-grained actions, and spatial relationships.

Figure 7: Local guidance prompt to guide the token compression if conversation history exists. "last_conv" refers to the last user query and the corresponding model answer from the conversation history.

> Describe the current scene in detail, focusing on specific objects, fine-grained actions, and spatial relationships.

Figure 8: Local guidance prompt to guide the token compression if there is no conversation history.

> Context summary: {last_conv}. Summarize the video narrative, identifying main characters, key events, timeline changes, and the overall theme.

Figure 9: Global guidance prompt to guide the token compression if conversation history exists. "last_conv" refers to the last user query and the corresponding model answer from the conversation history.

## D Configuration of Cross-Layer Memory Smoothing

Given that long-term memory tends to remain relatively stable, while short-term memory focuses on diverse perception, we set different $\lambda$ for different layer stages:

$$\lambda_l = \begin{cases} 0.1, & \text{if } l \in \mathcal{L}_{shallow} \\ 0.3, & \text{if } l \in \mathcal{L}_{middle} \\ 0.4, & \text{if } l \in \mathcal{L}_{deep} \end{cases} \tag{7}$$

The ablation study Tab. 6 shows the effectiveness of this hyperparameter choice.

> Summarize the video narrative, identifying main characters, key events, timeline changes, and the overall theme.

Figure 10: Global guidance prompt to guide the token compression if there is no conversation history.

## E Details of evaluated benchmarks

### E.1 Streaming Benchmarks

- **StreamingBench** (Lin et al., 2024b) assesses the streaming video understanding capabilities of MLLMs. It evaluates three core aspects: real-time visual understanding, omni-source understanding, and contextual understanding. The Real-Time Visual Understanding subset is the most extensive component, featuring 2,500 questions across 500 videos. It covers 10 tasks, such as object perception and causal reasoning. In this paper, we focus on the Real-Time Visual Understanding subset for evaluation.
- **OVO-Bench** (Li et al., 2025) evaluates the online reasoning and temporal awareness of MLLMs, featuring 644 videos with approximately 2,800 fine-grained multiple-choice QA pairs. It organizes 12 tasks into three distinct categories, which are real-time visual perception, backward tracing, and forward active responding. Given that we do not focus on the proactive responding ability of MLLMs in this paper, we exclusively utilize the real-time perception and the backward tracing subsets.
- **RVS-Ego** and **RVS-Movie** (Zhang et al., 2024a) are designed to evaluate the real-time understanding capabilities of models in online streaming scenarios. The datasets consist of 10 long egocentric videos from the Ego4D dataset (Grauman et al., 2022) and 22 long movie clips from the MovieNet dataset (Huang et al., 2020) dataset, totaling over 21 hours of video content.

Table 11: **Key statistics of the streaming benchmarks.** In the "Type" column, "MC" denotes multiple-choice questions, while "OE" denotes open-ended questions. In the "Benchmark" column, "rt" denotes real-time understanding subset, while "bw" denotes backward tracing subset.

| Benchmark | Duration | #Videos | #QA | Type |
|---|---|---|---|---|
| StreamingBench$_{rt}$ | 10.1min | 500 | 2,500 | MC |
| OVO-Bench$_{bw}$ | 5.9 min | 275 | 631 | MC |
| OVO-Bench$_{rt}$ | 8.8 min | 237 | 837 | MC |
| RVS-Ego | 60 min | 10 | 1,465 | OE |
| RVS-Movie | 30 min | 22 | 1,905 | OE |

### E.2 Offline Benchmarks

- **MVBench** (Li et al., 2024b) systematically evaluates the temporal understanding capabilities of MLLMs. It utilizes a novel static-to-dynamic method to define 20 distinct temporal tasks, such as action sequence and moving direction, which cannot be effectively solved with a single frame. The videos are collected from a wide range of datasets, including NTU RGB+D (Shahroudy et al., 2016), Perception (Pătrăucean et al., 2023), etc.
- **Egoschema** (Mangalam et al., 2023) is a diagnostic benchmark designed to assess long-form video understanding abilities. Derived from Ego4D (Grauman et al., 2022), it consists of over 5,000 human-curated multiple-choice QA pairs associated with egocentric video clips.
- **VideoMME** (Fu et al., 2025) is a full-spectrum, multimodal benchmark designed for the comprehensive evaluation of MLLMs in video analysis. It comprises 900 manually curated videos spanning six primary domains and diverse durations to assess temporal adaptability. The dataset features 2,700 high-quality QA pairs that necessitate processing multimodal inputs, including video frames, subtitles, and audio.

Table 12: **Key statistics of the offline benchmarks.** In the "Type" column, "MC" denotes multiple-choice questions.

| Benchmark | Duration | #Videos | #QA | Type |
|---|---|---|---|---|
| MVBench | 16 s | 3,641 | 4,000 | MC |
| Egoschema | 3 min | 5,063 | 5,063 | MC |
| VideoMME | 17 min | 900 | 2,700 | MC |

## F Details of Position Re-Indexing

Inspired by StreamingVLM's strategy of managing positional stability in streaming scenarios (Xu et al., 2025), we adopt a unified left-compaction re-indexing scheme to eliminate positional gaps introduced by KV-cache pruning while preserving the semantic anchoring of the system prompt. Concretely, system text tokens are kept fixed to provide a stable textual anchor, whereas retained video tokens are re-indexed in a left-compact manner and placed contiguously after the static prefix. To reuse cached key states without re-computation, we further apply a delta-based rotary correction that compensates for the positional displacement.

### F.1 Re-indexing for LLaVA-OV (1D RoPE)

LLaVA-OV employs standard 1D RoPE, where each token is associated with a scalar positional index $p$. Therefore, we perform left-compaction of the 1D indices: the system prefix positions remain unchanged, while the retained positions of video tokens are reassigned to form a dense contiguous segment immediately following the fixed prefix.

Let `offset` denote the length of the system prompt prefix tokens, and let

$$\mathcal{P} = \{p_0 < p_1 < \cdots < p_{N-1}\}$$

be the sorted set of retained video token positions (excluding the fixed prefix). For a retained video token originally at position $p_{\text{old}} \in \mathcal{P}$, its compacted 1D position is defined as

$$p_{\text{new}} = \texttt{offset} + \text{rank}_{\mathcal{P}}(p_{\text{old}})\,. \tag{8}$$

This mapping removes gaps while preserving the original temporal ordering along the stream, and ensures that the video region occupies a dense range directly after the static text region.

To align cached key states with the updated positions, we avoid re-generating keys and instead apply a rotary delta correction induced by the positional shift. For a cached key vector $\mathbf{k}_{\text{old}}$ associated with position $p_{\text{old}}$ and remapped to $p_{\text{new}}$, we compute

$$\mathbf{k}_{\text{new}} = \mathbf{k}_{\text{old}} \odot \text{RotaryDelta}(p_{\text{old}}, p_{\text{new}})\,, \tag{9}$$

where the relative phase shift is

$$\text{RotaryDelta}(p_{\text{old}}, p_{\text{new}}) = e^{i(p_{\text{new}} - p_{\text{old}})\boldsymbol{\theta}}, \tag{10}$$

and $\boldsymbol{\theta}$ denotes the RoPE frequency vector. This update preserves the correctness of attention under the new indexing while enabling direct reuse of the cached KV states.

### F.2 Re-indexing for Qwen2.5-VL (3D M-RoPE)

For Qwen2.5-VL, video tokens are indexed by a 3D M-RoPE coordinate $\mathbf{p} = (p^{(t)}, p^{(h)}, p^{(w)})$, covering temporal and spatial dimensions. After pruning, the retained video tokens typically occupy sparse coordinates along each dimension $d \in \{t, h, w\}$. To eliminate the gaps without disturbing the monotonic ordering, we apply dimension-wise left-compaction independently along each axis, while keeping the system token prefix fixed.

Let

$$\mathcal{P}^{(d)} = \{p_0^{(d)} < p_1^{(d)} < \cdots < p_{N_d-1}^{(d)}\}$$

denote the sorted set of retained coordinates along dimension $d$. For a token originally located at $p_{\text{old}}^{(d)} \in \mathcal{P}^{(d)}$, its compacted coordinate is defined by its rank within $\mathcal{P}^{(d)}$, shifted by the fixed prefix `offset`:

$$p_{\text{new}}^{(d)} = \texttt{offset} + \text{rank}_{\mathcal{P}^{(d)}}\left(p_{\text{old}}^{(d)}\right), d \in \{t, h, w\}. \tag{11}$$

This procedure yields a dense and contiguous $(t, h, w)$ grid for the video tokens placed immediately after the static text region, thereby ensuring positional continuity while preserving the distinct semantic roles of temporal and spatial indices.

As in the 1D case, we reuse cached keys by applying a M-RoPE correction. Given a key $\mathbf{k}_{\text{old}}$ associated with

$$\mathbf{p}_{\text{old}} = (p_{\text{old}}^{(t)}, p_{\text{old}}^{(h)}, p_{\text{old}}^{(w)})$$

and remapped to

$$\mathbf{p}_{\text{new}} = (p_{\text{new}}^{(t)}, p_{\text{new}}^{(h)}, p_{\text{new}}^{(w)}),$$

the corrected key is obtained as

$$\mathbf{k}_{\text{new}} = \mathbf{k}_{\text{old}} \odot \text{RotaryDelta}(\mathbf{p}_{\text{old}}, \mathbf{p}_{\text{new}}), \tag{12}$$

with the relative phase shift:

$$\begin{aligned} &\text{RotaryDelta}(\mathbf{p}_{\text{old}}, \mathbf{p}_{\text{new}}) \\ &= \underset{d \in \{t,h,w\}}{\text{Concat}} \left(e^{i(p_{\text{new}}^{(d)} - p_{\text{old}}^{(d)})\boldsymbol{\theta}^{(d)}}\right), \end{aligned} \tag{13}$$

where Concat denotes the concatenation operation along the channel dimension, and $\boldsymbol{\theta}^{(d)}$ represents the rotary frequency vector corresponding to the channel section allocated for dimension $d$.

## G Algorithm of Summary Tokens

**Algorithm 1** Summary Token Aggregation

**Require:** $K_p, V_p$: Pruned KV tensors from visual tokens; $P_p$: Original position indices of pruned tokens; $t$: Target position index for the summary token.

**Ensure:** $k_{sum}, v_{sum}$: Single aggregated summary token cache.

**Step 1: Aggregate Value**
# Simple spatial mean
$v_{sum} \leftarrow \text{Mean}(V_p)$

**Step 2: Aggregate Key**
# Phase alignment before pooling
$\Delta\theta \leftarrow \text{RotaryDelta}(P_p \rightarrow t)$
# Calculate rotation shift from $P_p$ to $t$
$K_{aligned} \leftarrow \text{ApplyDelta}(K_p, \Delta\theta)$
# Align all keys to the same phase
$k_{sum} \leftarrow \text{Mean}(K_{aligned})$

**Step 3: Update KV Cache**
$K_{new} \leftarrow \text{Concat}([K_{kept}, k_{sum}])$
$V_{new} \leftarrow \text{Concat}([V_{kept}, v_{sum}])$

**return** $K_{new}, V_{new}$

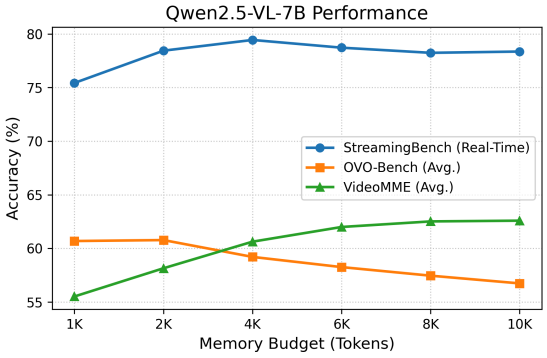



Figure 11: Performance comparison of Qwen2.5-VL-7B across different memory budgets.

## H More Ablation on Memory Budget

Here we demonstrate the ablation study on memory budget using HERMES built on Qwen2.5-VL-7B, as shown in Fig. 11. The conclusions regarding memory budget on Qwen2.5-VL-7B is consistent with those observed on LLaVA-OV-7B, which is reported in Sec. 4.4.

## I Full Performances

### I.1 StreamingBench

### I.2 OVO-Bench

## J Case Study

We provide six representative case study examples from RVS-Ego and RVS-Movie to demonstrate the advantages of HERMES compared to the foundation model LLaVA-OV-7B. During the under-

Table 13: **Accuracy comparison (%) on StreamingBench focusing on *Real-Time Visual Understanding* tasks**. Real-Time Visual Understanding tasks consists of Object Perception (OP), Causal Reasoning (CR), Clips Summarization (CS), Attribute Perception (ATP), Event Understanding (EU), Text-Rich Understanding (TR), Prospective Reasoning (PR), Spatial Understanding (SU), Action Perception (ACP), and Counting (CT).

| Model | #Frames | OP | CR | CS | ATP | EU | TR | PR | SU | ACP | CT | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Human | - | 89.47 | 92.00 | 93.60 | 91.47 | 95.65 | 92.52 | 88.00 | 88.75 | 89.74 | 91.30 | 91.46 |
| **Proprietary MLLMs** | | | | | | | | | | | | |
| Gemini 1.5 pro (Comanici et al., 2025) | 1 fps | 79.02 | 80.47 | 83.54 | 79.67 | 80.00 | 84.74 | 77.78 | 64.23 | 71.95 | 48.70 | 75.69 |
| GPT-4o (OpenAI et al., 2024) | 64 | 77.11 | 80.47 | 83.91 | 76.47 | 70.19 | 83.80 | 66.67 | 62.19 | 69.12 | 49.22 | 73.28 |
| Claude 3.5 Sonnet (Anthropic, 2024) | 20 | 73.33 | 80.47 | 84.09 | 82.02 | 75.39 | 79.53 | 61.11 | 61.79 | 69.32 | 43.09 | 72.44 |
| **Open-source Offline MLLMs** | | | | | | | | | | | | |
| Video-LLaMA2-7B (Cheng et al., 2024) | 32 | 55.86 | 55.47 | 57.41 | 58.17 | 52.80 | 43.61 | 39.81 | 42.68 | 45.61 | 35.23 | 49.52 |
| VILA-1.5-8B (Lin et al., 2024a) | 14 | 53.68 | 49.22 | 70.98 | 56.86 | 53.42 | 53.89 | 54.63 | 48.78 | 50.14 | 17.62 | 52.32 |
| Video-CCAM-14B (Fei et al., 2024) | 96 | 56.40 | 57.81 | 65.30 | 62.75 | 64.60 | 51.40 | 42.59 | 47.97 | 49.58 | 31.61 | 53.96 |
| LongVA-7B (Zhang et al., 2024c) | 128 | 70.03 | 63.28 | 61.20 | 70.92 | 62.73 | 59.50 | 61.11 | 53.66 | 54.67 | 34.72 | 59.96 |
| InternVL-V2-8B (Chen et al., 2024b) | 16 | 68.12 | 60.94 | 69.40 | 77.12 | 67.70 | 62.93 | 59.26 | 53.25 | 54.96 | 56.48 | 63.72 |
| Kangaroo-7B (Liu et al., 2024b) | 64 | 71.12 | 84.38 | 70.66 | 73.20 | 67.08 | 61.68 | 56.48 | 55.69 | 62.04 | 38.86 | 64.60 |
| LLaVA-NeXT-Video-32B (Liu et al., 2024a) | 64 | 78.20 | 70.31 | 73.82 | 76.80 | 63.35 | 69.78 | 57.41 | 56.10 | 64.31 | 38.86 | 66.96 |
| MiniCPM-V-2.6-8B (Hu et al., 2024) | 32 | 71.93 | 71.09 | 77.92 | 75.82 | 64.60 | 65.73 | 70.37 | 56.10 | 62.32 | 53.37 | 67.44 |
| **Open-source Online MLLMs** | | | | | | | | | | | | |
| Flash-VStream-7B (Zhang et al., 2024b) | 1 fps | 25.89 | 43.57 | 24.91 | 23.87 | 27.33 | 13.08 | 18.52 | 25.20 | 23.87 | 48.70 | 23.23 |
| VideoLLM-online-8B (Chen et al., 2024a) | 2 fps | 39.07 | 40.06 | 34.49 | 31.05 | 45.96 | 32.40 | 31.48 | 34.16 | 42.49 | 27.89 | 35.99 |
| Dispider-7B (Qian et al., 2025) | 1 fps | 74.92 | 75.53 | 74.10 | 73.08 | 74.44 | 59.92 | 76.14 | 62.91 | 62.16 | 45.80 | 67.63 |
| TimeChat-Online-7B (Yao et al., 2025) | 1 fps | 80.22 | 82.03 | 79.50 | 83.33 | 76.10 | 78.50 | 78.70 | 64.63 | 69.60 | 57.98 | 75.36 |
| StreamForest-7B (Zeng et al., 2025) | 1 fps | 83.11 | 82.81 | 82.65 | 84.26 | 77.50 | 78.19 | 76.85 | 69.11 | 75.64 | 54.40 | 77.26 |
| **Training-free Offline-to-Online Methods** | | | | | | | | | | | | |
| LLaVA-OV-7B (Li et al., 2024a) | 32 | 78.75 | 78.12 | 80.76 | **81.19** | 71.70 | 72.59 | 72.22 | 63.82 | 66.01 | 38.34 | 71.34 |
| + ReKV (Di et al., 2025) | 0.5 fps | 76.02 | 81.25 | 77.92 | 76.90 | 66.04 | 66.04 | 69.44 | 60.98 | 64.31 | 49.22 | 69.22 |
| + LiveVLM (Ning et al., 2025) | 0.5 fps | **81.47** | 78.13 | 83.28 | 79.08 | 69.57 | **74.14** | **75.00** | **69.11** | 67.71 | 40.41 | 72.92 |
| + StreamKV (Chen et al., 2025b) | 0.5 fps | 73.80 | 77.30 | 85.90 | 77.50 | **73.30** | 63.90 | 69.40 | 61.40 | 63.20 | 35.80 | 68.80 |
| + HERMES (6K tokens) | 0.5 fps | 77.93 | **82.03** | 86.12 | **81.19** | 66.04 | 73.52 | 74.07 | 63.01 | 67.71 | **45.08** | 72.63 |
| + HERMES (4K tokens) | 0.5 fps | 79.02 | 81.25 | **87.70** | 80.20 | 69.18 | 71.96 | 73.15 | 66.26 | **69.41** | 43.52 | **73.23** |
| LLaVA-OV-0.5B (Li et al., 2024a) | 32 | 71.39 | 57.81 | 65.93 | 69.64 | 69.18 | 55.76 | 57.41 | **52.85** | 62.04 | 16.58 | 59.64 |
| + ReKV (Di et al., 2025) | 0.5 fps | 65.12 | 60.16 | 66.56 | 66.01 | 66.67 | 52.96 | 57.41 | 48.37 | 60.34 | 18.13 | 57.39 |
| + HERMES (6K tokens) | 0.5 fps | 71.93 | 60.16 | 69.09 | 71.29 | 68.55 | 57.32 | **60.19** | 51.22 | **63.74** | **19,69** | 61.04 |
| + HERMES (4K tokens) | 0.5 fps | **72.21** | **61.72** | **70.98** | **72.94** | **72.33** | **57.94** | **60.19** | **52.85** | **63.74** | 19.17 | **62.04** |
| Qwen2.5-VL-7B (Bai et al., 2025b) | 1 fps | 77.93 | 76.56 | 78.55 | 80.86 | 76.73 | 76.95 | 80.56 | 65.45 | 65.72 | **52.85** | 73.31 |
| + HERMES (6K tokens) | 0.5 fps | 83.38 | 78.91 | 86.12 | 87.13 | **78.62** | **86.60** | **84.26** | 74.80 | 71.39 | 46.63 | 78.72 |
| + HERMES (4K tokens) | 0.5 fps | **83.65** | **81.25** | **88.01** | **87.46** | 76.73 | **86.60** | 82.41 | **76.02** | **73.94** | 46.63 | **79.44** |
| Qwen2.5-VL-32B (Bai et al., 2025b) | 1 fps | 76.29 | 79.69 | 78.55 | **83.50** | 76.10 | 79.44 | 80.56 | 61.38 | 68.27 | **59.07** | 74.27 |
| + HERMES (6K tokens) | 0.5 fps | **84.47** | 79.69 | **87.70** | 83.17 | **81.76** | **88.16** | 86.11 | 74.80 | **77.62** | 49.22 | **80.20** |
| + HERMES (4K tokens) | 0.5 fps | 83.92 | **80.47** | **87.70** | **83.50** | 80.50 | **88.16** | **87.04** | **75.20** | 77.34 | 48.19 | 80.08 |

standing of streaming long videos, HERMES exhibits significantly finer-grained temporal (shown in Fig. 12) and spatial understanding Fig. 13 capabilities than its corresponding foundation model.

Table 14: **Accuracy comparison (%) on OVO-Bench focusing on *Real-Time Visual Perception* and *Backward Tracing* tasks**. Real-Time Visual Perception tasks consist of Optical Character Recognition (OCR), Action Recognition (ACR), Attribute Recognition (ATR), Spatial Understanding (STU), Future Prediction (FPD), Object Recognition (OJR). Backward Tracing tasks consists of Episodic Memory (EPM), Action Sequence Identification (ASI), Hallucination Detection (HLD).

| Model | #Frames | Real-Time Visual Perception | | | | | | | Backward Tracing | | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | OCR | ACR | ATR | STU | FPD | OJR | Avg. | EPM | ASI | HLD | Avg. | Avg. |
| Human | - | 93.96 | 92.57 | 94.83 | 92.70 | 91.09 | 94.02 | 93.20 | 92.59 | 93.02 | 91.37 | 92.33 | 92.77 |
| **Proprietary MLLMs** | | | | | | | | | | | | | |
| Gemini 1.5 Pro (Comanici et al., 2025) | 1 fps | 85.91 | 66.97 | 79.31 | 58.43 | 63.37 | 61.96 | 69.32 | 58.59 | 76.35 | 52.64 | 62.54 | 65.93 |
| GPT-4o (OpenAI et al., 2024) | 64 | 69.80 | 64.22 | 71.55 | 51.12 | 70.30 | 59.78 | 64.46 | 57.91 | 75.68 | 48.66 | 60.75 | 62.61 |
| **Open-source Offline MLLMs** | | | | | | | | | | | | | |
| LLaVA-Video-7B (Zhang et al., 2025) | 64 | 69.80 | 59.63 | 66.38 | 50.56 | 72.28 | 61.41 | 63.34 | 51.18 | 64.19 | 9.68 | 41.68 | 52.51 |
| Qwen2-VL-7B (Wang et al., 2024b) | 64 | 69.13 | 53.21 | 63.79 | 50.56 | 66.34 | 60.87 | 60.65 | 44.44 | 66.89 | 34.41 | 48.58 | 54.62 |
| InternVL2-8B (Chen et al., 2024b) | 64 | 68.46 | 58.72 | 68.97 | 44.94 | 67.33 | 55.98 | 60.73 | 43.10 | 61.49 | 27.41 | 44.00 | 52.37 |
| LongVU-7B (Shen et al., 2024) | 1 fps | 55.70 | 49.54 | 59.48 | 48.31 | 68.32 | 63.04 | 57.40 | 43.10 | 66.22 | 9.14 | 39.49 | 48.45 |
| **Open-source Online MLLMs** | | | | | | | | | | | | | |
| VideoLLM-online-8B (Chen et al., 2024a) | 2 fps | 8.05 | 23.85 | 12.07 | 14.04 | 45.54 | 21.20 | 20.79 | 22.22 | 18.80 | 12.18 | 17.73 | 19.26 |
| Flash-VStream-7B (Zhang et al., 2024b) | 1 fps | 25.50 | 32.11 | 29.31 | 33.71 | 29.70 | 28.80 | 29.86 | 36.36 | 33.78 | 5.91 | 25.35 | 27.61 |
| Dispider-7B (Qian et al., 2025) | 1 fps | 57.72 | 49.54 | 62.07 | 44.94 | 61.39 | 51.63 | 54.55 | 48.48 | 55.41 | 4.30 | 36.06 | 45.31 |
| TimeChat-Online-7B (Yao et al., 2025) | 1 fps | 75.20 | 46.80 | 70.70 | 47.80 | 69.30 | 61.40 | 61.90 | 55.90 | 59.50 | 9.70 | 41.70 | 51.80 |
| StreamForest-7B (Zeng et al., 2025) | 1 fps | 68.46 | 53.21 | 71.55 | 47.75 | 65.35 | 60.87 | 61.20 | 58.92 | 64.86 | 32.26 | 52.02 | 56.61 |
| **Training-free Offline-to-Online Methods** | | | | | | | | | | | | | |
| LLaVA-OV-7B (Li et al., 2024a) | 32 | 67.79 | 55.05 | 72.41 | 48.31 | 72.28 | 62.50 | 63.06 | 57.24 | 55.41 | 18.28 | 43.64 | 53.35 |
| + ReKV (Di et al., 2025) | 0.5 fps | 52.35 | 54.13 | 69.83 | 43.26 | 67.33 | 57.07 | 57.33 | 57.58 | 56.08 | 18.82 | 44.16 | 50.75 |
| + HERMES (6K tokens) | 0.5 fps | **72.48** | **62.39** | 69.83 | 47.75 | **73.27** | 64.67 | 65.07 | **61.28** | 58.78 | 26.34 | 48.80 | 56.94 |
| + HERMES (4K tokens) | 0.5 fps | **72.48** | **62.39** | **74.14** | **50.56** | **73.27** | **65.22** | **66.34** | 60.61 | **61.49** | **28.49** | **50.20** | **58.27** |
| LLaVA-OV-0.5B (Li et al., 2024a) | 32.00 | 53.69 | 53.21 | 48.28 | **33.71** | 60.40 | **48.91** | 49.70 | 46.13 | 45.27 | **12.37** | 34.59 | 42.15 |
| + ReKV (Di et al., 2025) | 0.5 fps | 41.61 | 44.95 | 50.00 | 29.78 | 60.40 | 35.87 | 43.77 | 46.13 | 43.92 | 9.14 | 33.06 | 38.42 |
| + HERMES (6K tokens) | 0.5 fps | **57.05** | **49.54** | 55.17 | 32.58 | 60.40 | 47.28 | 50.34 | **47.81** | 47.30 | 9.14 | 34.75 | 42.55 |
| + HERMES (4K tokens) | 0.5 fps | 56.38 | 47.71 | **56.90** | 32.02 | **62.38** | **48.91** | **50.72** | **47.81** | **47.97** | 8.60 | **34.80** | **42.76** |
| Qwen2.5-VL-7B (Bai et al., 2025b) | 1 fps | 67.79 | 55.05 | 67.24 | 42.13 | 66.34 | 60.87 | 59.90 | **51.52** | 58.78 | 23.66 | 44.65 | 52.28 |
| + HERMES (6K tokens) | 0.5 fps | **85.91** | 60.55 | **74.14** | 52.81 | 70.30 | **66.85** | 68.42 | 49.49 | 61.49 | 33.33 | 48.10 | 58.26 |
| + HERMES (4K tokens) | 0.5 fps | 85.23 | **64.22** | 71.55 | **53.37** | **74.26** | 65.22 | **68.98** | 48.48 | **62.16** | **37.63** | **49.43** | **59.21** |
| Qwen2.5-VL-32B (Bai et al., 2025b) | 1 fps | 77.18 | 58.72 | 68.10 | 50.56 | **74.26** | 57.61 | 64.40 | **58.59** | 62.84 | 29.57 | 50.33 | 57.37 |
| + HERMES (6K tokens) | 0.5 fps | 87.25 | **66.06** | **74.14** | 57.30 | 71.29 | 75.54 | 71.93 | 55.56 | **70.27** | 47.31 | **57.71** | **64.82** |
| + HERMES (4K tokens) | 0.5 fps | **88.59** | 65.14 | **74.14** | **58.99** | 71.29 | **76.09** | **72.37** | 52.19 | 66.22 | **47.85** | 55.42 | 63.90 |

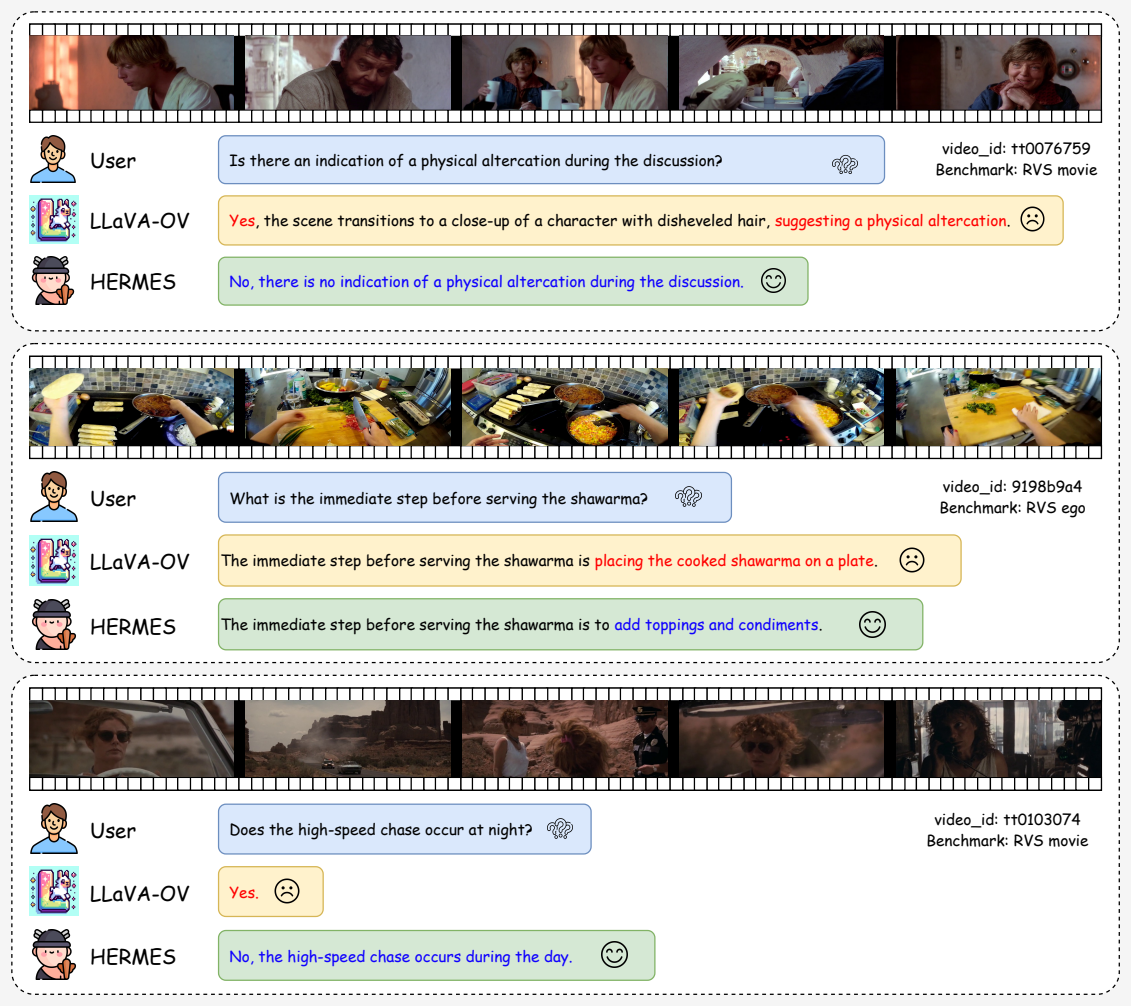



Figure 12: Cases demonstrating the superior fine-grained temporal understanding capability of HERMES relative to the LLaVA-OV-7B base model.

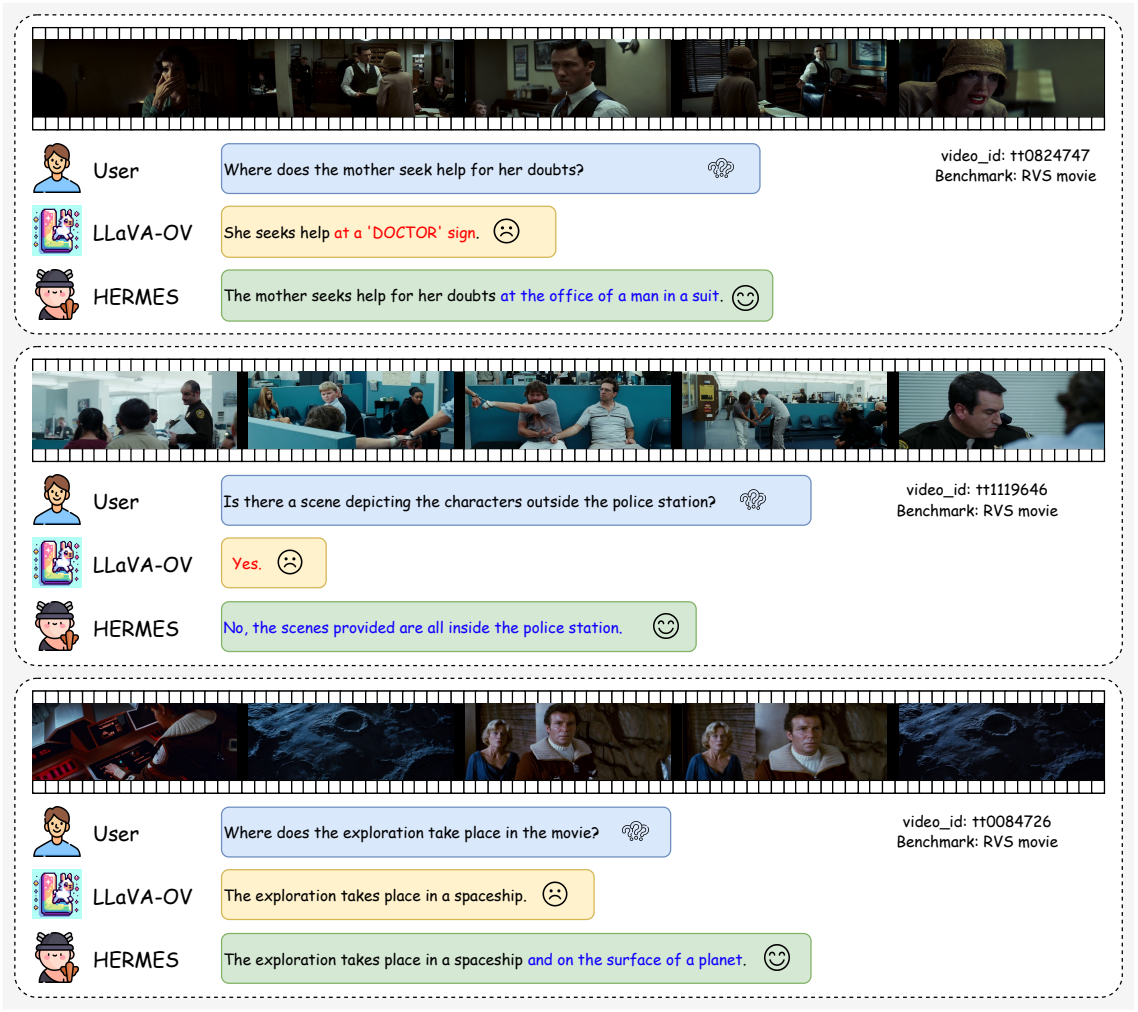



Figure 13: Cases demonstrating the superior fine-grained spatial understanding capability of HERMES relative to the LLaVA-OV-7B base model.